UNIVERSIDADE FEDERAL DE SANTA CATARINA

DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

BACHARELADO EM CIÊNCIAS DA COMPUTAÇÃO

Implementação de um controlador para a Virtual Robots Competitions da RoboCup

Marcelo Ribeiro Xavier da Silva

Trabalho de conclusão de curso apresentada como parte dos requisito para obtenção do grau Bacharel em Ciências da Computação

Florianópolis – SC

2007/2

Marcelo Ribeiro Xavier da Silva

Implementação de um controlador para a Virtual Robots Competitions da RoboCup

Trabalho de conclusão de curso apresentada como parte dos requisito para obtenção do grau Bacharel em Ciências da Computação

Orientador: Professor Doutor Mauro Roisenberg

Banca Examinadora

____

Professor Doutor Jovelino Falqueto

____

Professor Doutor Ricardo Azambuja Silveira

Sumário

Lista de Figuras

1 *Introdução*

A inteligência artificial é a área de pesquisa das ciências da computação dedicada a "imitar"a capacidade humana de resolver problemas, ou seja, através de métodos ou algoritmos computacionais tenta simular inteligência fazendo com que o computador resolva problemas através do aprendizado. Ou ainda, segundo Elaine Rich(RICH; KNIGHT, 1994), "A inteligência Artificial (I.A.) é o estudo de como fazer os computadores realizarem tarefas em que, no momento, as pessoas são melhores".

Segundo J. Norberto Pires(PIRES, 2002), a palavra robô, é derivada de robot que vem do checo e significa trabalho. Ou seja, um robô é uma máquina que exerce um trabalho específico, um trabalho para o qual fora justamente criada. Com a injeção de Inteligência Artificial, essas máquinas estão exercendo cada vez melhor suas funções e estão se tornando cada vez mais abrangentes, executando cada vez mais tarefas.

Com a finalidade de promover a Inteligência Artificial e a Robótica Inteligente foi criada uma junta Internacional, chamada de RoboCup(ROBOCUP,). Para cumprir essa função de disseminar e fazer evoluir a cultura da robótica, a RoboCup promove várias competições que objetivam a integração de inúmeras tecnologias, algoritmos e métodos. Dentre estas competições, existe a Virtual Robots Competition(VIRTUAL. . . ,) que, como induz o nome, é uma simulação de robôs.

A Virtual Robots Competition introduz um problema de busca e reconhecimento de padrões. Dentro desta temática, é criada uma equipe robótica que deve procurar e pré-diagnosticar feridos em um mapa previamente disponibilizado. Ou seja, é a simulação dessa equipe robótica de busca que num mapa, disponibilizado pelos organizadores, tenta encontrar possíveis feridos, dando um pré-diagnóstico do estado de cada vítima. Esta competição usa o Urban Search And Rescue Simulation (USARSim)(USARSIM,), um simulador de alta fidelidade que usa o engine do jogo UnrealTournament (comprado separadamente). Dentro do USARSim são gerados os mapas, os robôs, as vítimas, os obstáculos, toda a simulação dos sensores semelhantes aos reais e toda a interação desses objetos, dessa forma os usuários podem simular os múltiplos agentes,

cujas potencialidades se assemelham àquelas dos robôs reais. Todas as regras(VIRTUAL...,), incluindo sensores permitidos, podem ser encontradas no site da competição.

Para fazer a inserção dos robôs no simulador, existem ferramentas chamadas de controladores. Porém estas ferramentas não servem apenas para inserir os robôs no simulador. Como o nome leva a pensar, elas servem para fazer o controle de toda a informação gerada e/ou lida pela equipe de robôs, ou seja, servem para fazer toda interação dos usuários com os sensores dos robôs. No manual de instalação do USARSim é citado o Mobility Open Architecture Simulation and Tools (MOAST)(MOAST,) que é um dos controladores suportados pelo simulador. O manual do USARSim cita ainda dois outros controladores, porém, a utilização destes controladores não é intuitiva, além do fato de ser necessário ter profundo conhecimento sobre como fazer a entrada dos dados e a leitura de informações devolvidas por eles, tornando-os não muito didáticos.

Considerando todos estes fatores que dificultam a utilização dos controladores para a Virtual Robots Competition, pesquisadores do Instituto Tecnológico da Aeronáutica (ITA) desenvolveram um controlador em C++, que deveria, além de servir para a competição, suprir a necessidade de uma ferramenta eficiente para ensino de robótica e inteligência artificial. Para que esta ferramenta fosse eficiente ela deveria ser tão intuitiva que não prenderia o aluno no funcionamento dela em si, mas sim no seu objetivo ao utilizá-la. O controlador criado pelo ITA já pode ser usado na competição, porém ainda não está completo para ser utilizado como ferramenta didática. Portanto, em parceria com eles, e com outras instituições, pretendemos finalizar este processo criativo.

1.1 Justificativa

Existem inúmeros controladores para a Virtual Robots Competitions da RoboCup, porém nenhum deles foi considerado satisfatório por não terem em suas interfaces uma forma simples para entrada e saída de dados, tornando o uso complicado e nada intuitivo.

A idéia inicial dessa parceria entre universidades é criar um controlador que seja tão simples de ser usado que venha a ser uma ferramenta para o aprendizado de conceitos e técnicas de Inteligência Artificial.

Acreditamos que ao criar um controlador simples em termo de uso e completo em termos de entrada e saída de dados, não apenas irá ajudar a aumentar o interesse das pessoas na área de Inteligência Artificial como também irá aumentar a participação de brasileiros na Virtual Robots Competitions.

O ideal deste trabalho é ajudar a longo prazo a desenvolver a área de Inteliência Artificial, criando e/ou aumentando o interesse entre os estudantes brasileiros tanto na área quanto em compentições que busquem o mesmo objetivo.

1.2 Objetivos

1.2.1 Objetivos Gerais

O objetivo deste trabalho é criar um controlador cuja implementação seja inovadora e cumpra as regras definidas na RoboCup para a Virtual Robots Competitions. Este controlador será usado como ferramenta didática para estudo de técnicas de Inteligência Artificial, para isso, deve ser bastante intuitivo e fácil de usar.

1.2.2 Objetivos específicos

Para alcançar os objetivos gerais definidos para o trabalho, os seguintes objetivos específicos foram listados:

- Fazer com que o controlador seja adaptado às regras da Virtual Robots Competitions
- Adaptar o controlador, tornando-o:
  - Multiplataforma
  - Extensível
  - Escalável
  - Fácil para testar novos algoritmos

1.3 Metodologia

O trabalho foi realizado e está organizado da seguinte maneira:

No capítulo 2 - É explicada a Virtual Robots Competitions.

Para explicar o funcionamento da Virtual Robots Competitions foi usado como base o documento, disponibilizado pelos organizadores do evento, que contêm as regras da competição.

Após a análise das regras da competição, o modelo de controlador é proposto reunindo os requisitos que foram discutidos com as demais entidades envolvidas no processo criativo. A proposta é apresentada no capítulo 3.

O trabalho termina com as definições ainda pendentes e que necessitam da reunião com as demais entidades envolvidas para que sejam discutidas.

2 *Regras e funcionamento da Virtual Robots Competitions*

2.1 Escopo e Definição

A Virtual Robots Competition se dá em cima do simulador USARSim e é realizada dentro da RoboCupRescue Simulation league. Ela tem como escopo promover a troca de algoritmos entre a RoboCupRescue Robot League e a RoboCupRescue Simulation Agent competition(ROBOCUP. . . ,).

2.2 Configurações de Software e Hardware

Durante a competição existem dois sets de computadores (chamados de cluster). O simulador USARSim roda em modo cliente. Destes dois clusters, um é usado pelo time durante a competição, enquanto o outro é usado para que o time se prepare para a próxima competição. Sensores do tipo GroundTruth estão disponíveis preliminarmente à competição. Cada time roda seu código em máquinas próprias. Um cabo TCP/IP é provido para o time se conectar ao cluster. Todas as comunicações (operador-robô e robô-robô) usam o servidor de comunicação wireless (WCS). A figura abaixo representa o esquema de comunicação que é usado durante as competições. Para acessar o vídeo a estação base operadora deverá usar o WCS. Dessa forma, os vídeos serão obtidos apenas quando o robô mantiver contato com o rádio.

2.3 Esquema da Competição

A competição possui tanto testes de qualificação (ou testes elementares) quanto testes de cenário. A fim de participar nos cenários, as equipes terão de exibir competência em habilidades básicas como mapeamento e mobilidade no USARSim. Habilidades para operar nas zonas fáceis e intermediárias do ambiente de teste serão necessárias para ter sucesso nos cenários da

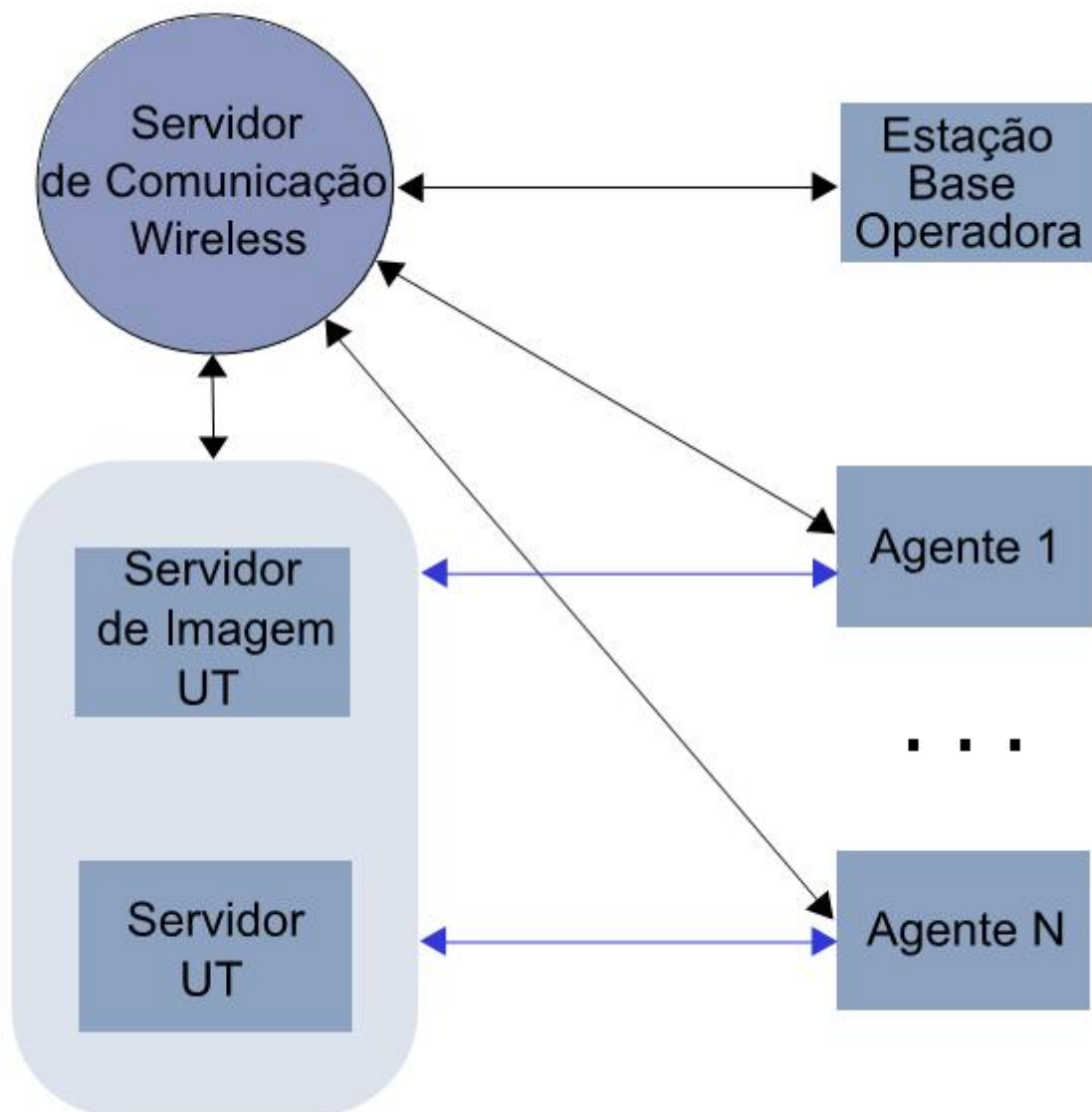



Figura 2.1: Wireless Comunication Server (VIRTUAL. . . ,)

competição. As áreas de maior dificuldade do ambiente de teste mostram o rumo que a competição tomará no próximo ano, e podem ser usadas para testar soluções avançadas e inovadoras.

2.4 Testes Elementares

Os testes elementares serão dividos em mapeamento e mobilidade.

2.4.1 Mobilidade

Para o teste de mobilidade um time de robôs será iniciado numa área prescrita. Para passa neste teste, pelo menos um robô deve atingir um determinado local-alvo na zona amarela antes de tempo expirar. Metas adicionais nas zonas laranja e vermelha serão colocadas para que os robôs tentem alcança-las. O tamanho da arena será comprimido de tal forma que a tarefa poder ser completada ou por um comportamento busca direta do alvo ou por exploração cooperativa (o alvo é alcançado durante a exploração). Durante estes testes estão disponíveis os sensores de relevo (GroundTruth) e os robôs tem a comunicação perfeita, ainda que por WCS.

2.4.2 Mapeamento/Localização de Vítima

Para o teste de mapeamento os robôs precisam mapear uma grande percentagem do mundo em que estão inseridos em um dado tempo. O mapa deve conter anotações de área explorada, localização das vítimas e o caminho traçado pelos robôs. A arena que é usada é similar à mobility arena e incorpora perfeita comunicação, apesar de haver ruídos nas leituras dos sensores, como os de odometria ou RFIds. O sensor GroundTruth também é permitido nesta etapa. Durante os testes, o time de robôs deve encontrar e relatar a localização de, pelo menos, uma vítima no tempo prescrito.

2.5 Cenários

Os cenários usados durante a competição são os cenários internos e externos do USAR. Um dia antes ao da execução, as equipes recebem informações básicas sobre o cenário. Dentre elas o tipo do cenário, interno ou externo, e informações disponíveis a priori. As informações que podem ser divulgadas a priori são do tipo localização de um evento, como o derramamento de um elemento químico em determinado ponto do mapa, ou simplesmente mapas do ambiente da execução. O formato do mapa é GeoTiff. Deve-se considerar que os mapas podem não estar atualizados e podem, portanto, conter erros devido às renovações, passagens bloqueadas, etc.

2.6 Antes da execução

Antes das execuções, cada time recebe um documento de texto que contêm as coordenadas da posição de início (x, y, z) e orientação para cada robô. As coordenadas de posicionamento são expressas em metros, e as orientações são em radianos. O arquivo consiste em número de

linhas iguais ao número de robôs, sendo que cada linha é composta pelo número dela seguido da coordenada do ponto de início e das orientações do robô a que ela se refere. Essas operações terão início trinta minutos antes da execução efetiva. Cada time irá se preparar no cluster que não está sendo utilizado. A execução começa no tempo programado. Independente da equipe não estar pronta o tempo começa a correr. Os robôs usam baterias que operam de 20 a 25 minutos. No tempo prescrito, os robôs devem ser inicializados dentro do simulador. Dentro do mapa os robôs devem aguardar por um comando de "start" que deve ser emitido antes que eles comecem a explorar a área. Quem monitora a condição da bateria, é o próprio robô. Informações não relatadas e colocadas no "log" antes da bateria expirar não são contabilizadas na contagem total dos pontos. Todos os robôs devem ser inseridos no mapa ao mesmo tempo, embora os times possam decidir ativá-los no momento que acharem mais conveniente.

2.7 Depois da execução

Para fins de pontuação cada equipe deve prover os seguintes elementos:

- Obrigatório: Um arquivo de texto simples, chamado de arquivo de vítimas, contendo as vítimas encontradas durante a exploração. Este arquivo de vítimas, deve conter a localização (x, y, z) de cada uma das vítimas em um referencial coerente com o mapa e as localizações de início dadas. O formato do arquivo de vítimas é dividido em linhas. Cada linha contêm, exatamente nesta ordem, o identificador da vítima (atribuído pela equipe), a posição em metros e a especificação de como a vítima foi encontrada. O último campo pode ter dois valores válidos. Se o sensor da vítima for usado para encontrá-la, então o valor atribuído ao campo é VSENSOR, porém, se não for usado esse sensor, o valor que o campo recebe é VISUAL. A última linha do arquivo de vítimas deve conter uma string com o valor END. O campo ID pode receber qualquer valor de string, pois serve, única e exclusivamente, para identificar a vítima descoberta pela equipe na posição relatada.

- Obrigatório: Um arquivo, chamado de arquivo bitmap, contendo o mapa e informações adicionais. Este arquivo deve conter múltiplos mapas sobrepostos. Todas as sobreposições devem estar também no formato GeoTiff ou Mif e devem cobrir uma região por completo. A sugestão é que os mapas de sobreposições representem probabilidades ou agrupamentos de mapa, ou áreas cobertas por cada robô. As sobreposições devem representar a seguinte informação com as dadas colorações:

  - Não atravessável - Preto RGB(0, 0, 0)

- Completo (atravessável, e livre de vítimas) - Verde RGB(0, 255, 0)
  - Incompleto (atravessável, porém não necessariamente livre de vítimas) - Branco RGB(255, 255, 255)
  - Inexplorado - Azul RGB (0, 0, 255)
  - Localização de Vítima - Vermelho (255, 0, 0). Locais podem ser indicados com uma cruz, um círculo ou um símbolo similar. O tamanho do símbolo deve ser de um tamanho que na escala do mapa não seja superior a um metro quadrado. Se uma vítima caminhando for encontrada, deve ser indicado que a mesma estava a pé e deve ser apontado o seu caminho.
  - Opcional: Qualquer informação recolhida durante a competição. Por exemplo, "screenshots" das vítimas, informações adicionais relativas às vítimas. Em caso de "screenshots" serem fornecidos, o arquivo deve ser nomeado consistentemente com os nomes utilizados no arquivo de vítimas, a fim de fazer uma associação é imediata.
  - Todas as informações devem ser apresentadas ao árbitro em, no mais tardar, 15 minutos após o prazo de término da execução. Informações atrasadas não são aceitas para fins de pontuação.

É importante deixar claro que mesmo que uma equipe rode com dez robôs, ao final apenas um arquivo deve ser enviado. O operador que iniciou os robôs deve preparar este arquivo. Qualquer outra pessoa que manipula os arquivos gerados pelos robôs será contabilizada como um operador adicional. Cabe à equipe de decidir se uma abrangente fundiu arquivo ou o melhor entre os arquivos individuais devem ser apresentados.

2.8 Pontuação

O sistema de pontuação baseia-se no critério de recompensar equipes que sejam capazes de explorar, de forma autônoma e segura, partes significativas do ambiente, proporcionando mapas de alta qualidade e demonstrando a capacidade de busca e localização de vítimas usando modelos robóticos consistentes. Ou seja, ganham pontos adicionais as equipes que mostrarem um desempenho superior ao esperado.

Fatores de Mérito:

- Um sensor especial de localização de vítima é disponibilizado para uso das equipes. Por este sensor ser utilizado, cinco pontos são atribuídos para equipe a cada vítima que for

corretamente localizada. Uma boa localização significa encontrar a vítima no recinto correto e em um raio de até um metro do ponto exato em que esta se encontra. Para vítimas que estejam se movimentando, a localização deve estar a um metro da trajetória traçada pela vítima. Cinco pontos adicionais são dados para quem prover uma imagem da vítima. A imagem deve ter uma qualidade que seja no mínimo suficiente para avaliar o estado da vítima. Se uma equipe opta pela utilização de processamento de imagem para detectar vítimas, então dez pontos são acrescidos por fornecimento de imagem e de informações de localização. Como fora dito anteriormente, a foto deve ter uma qualidade que possibilite uma avaliação do estado da vítima.

- Até 50 pontos são concedidos por exploração. As equipes são obrigadas a fornecerem mapas consistentes com formato formalmente descrito. Equipes que completam um certo número de metros quadrados do ambiente (os valores dependem do ambiente, e são atribuídos pelos árbitros) recebem 50 pontos. Para porcentagens mais baixas, o número de pontos sofre um decréscimo linear de zero a cinco pontos de pena (até a penalidade atingir o valor máximo dado pela exploração) por cada vítima não reportada que esteja numa área tida como completada (zona marcada com a cor verde no arquivo de bitmap).

- Até 50 pontos são concedidos por mapeamento. O mapeamento é julgado pelos seguintes critérios:

  - Qualidade de métrica – A precisão do arquivo de mapa comparado com o terreno real.

  - Fusão de multi-veículos – As equipes devem transformar todos os arquivos de mapa em um só arquivo. As equipes que incluem as informações de saída de vários robôs numa só têm um bônus na pontuação.

  - Atribuição - Uma das razões para gerar um mapa é para transmitir informações. Esta informação é frequentemente representada como atributos no mapa. Pontos são acrescidos por inclusão de informações sobre a localização de vítimas, a localização de obstáculos, e os caminhos que os robôs tomaram.

  - Agrupamento – A mais alta tarefa de mapeamento é reconhecer que elementos discretos no mapa constituem outros elementos. Por exemplo, o fato de várias paredes formarem um cômodo ou um conjunto particular de obstáculos ser um carro. Pontos de bônus são concedidos por notar tais grupos no mapa.

  - Qualidade de esqueleto – O esqueleto do mapa reduz um mapa complexo em um conjunto de locais conectados. Por exemplo, quando se representa um corredor com

um número elevado saídas, um esqueleto pode representar esse corredor como uma linha, e símbolos ao longo dessa linha podem representar as portas. Um mapa pode ser impreciso em termos de medidas (um corredor pode ser representado com 20 metros de comprimento, ao invés de 15 metros), mas pode ter um esqueleto preciso (existem três portas antes da sala com vítimas). A categoria permite que os juízes dêem pontos por quão preciso o esqueleto do mapa está representado.

  – Utilidade - Um dos principais objetivos de fornecer um mapa é criar a possibilidade de uma primeira resposta ao utilizar o mapa para determinar quais áreas foram completadas, onde os riscos podem ser localizados, e onde as vítimas foram capturadas.

Fatores de penalidade:

- Operadores: se a equipe usa N operadores, a pontuação será dividia por N+1. Cada equipe usa automaticamente um operador.

- Alarmes falsos: Para cada vítima que for incorretamente reportada é aplicada uma penalidade de cinco pontos. Um alarme falso é tido quando a vítima está posicionada a mais de um metro do local que fora relatado. No caso de uma vítima que esteja se movimentando isso se aplica à distância da sua trajetória.

- Choque: Cada vítima tocada por um robô irá incorrer num penalidade de cinco pontos. Se um robô toca uma vítima, a equipe é penalizada com cinco pontos, porém se o mesmo robô tocar a mesma vítima várias vezes, não é penalizado mais de uma vez. Ou seja, para sofrer mais de uma penalidade neste caso, o robô teria que acertar vítimas diferentes. Choques com as vítimas são detectados automaticamente pelo servidor e suas informações são guardadas no arquivo de log. Bater nas paredes ou outras estruturas não acarretam pena, no entanto algumas estruturas podem ser instáveis e bater poderia causar um colapso.

Operadores: Por questões de pontuação, um operador é contabilizado, logo que o membro da equipe:

- Inicia um robô, entra com os dados de ponto de partida, ou exerça qualquer operação que seja necessária para o sucesso do início da missão de resgate.

- Controla um robô de forma ativa.

- Pára um robô antes do término da execução. Para, por exemplo, prevenir um choque do robô com uma vítima.

- Estiver envolvido de qualquer forma com o processo de reconhecimento de vítima.

De acordo com o descrito acima, cada equipe deve ter pelo menos um operador para configurar os robôs.

2.9 Política de Open Source

É requisitado para equipe vencedora e para as equipes que apresentarem algum tipo de implementação considerada inovadora pelos árbitros, que disponibilizem imediatamente para os organizadores uma cópia completa e funcional dos softwares criados. Os softwares serão compartilhados em http://www.robocuprescue.org/wiki/index.php?title=Virtualrobots, dando os devidos créditos aos autores. O código fonte da competição realizada no ano passado pode ser encontrado nessa mesma página.

2.10 Combinações de sensores

Qualquer combinação dos sensores listados abaixo é permitida, porém, combinações não realísticas entre robôs e sensores podem ser vetadas. Por exemplo, um robô Zerg utilizando um sensor SickLMS.

- Encoder (codificador)
- GPS
- Hokuyo
- INS
- Odometry (odometria)
- RobotCamera (câmera robótica)
- SickLMS
- Sonar
- VictSensor (sensor de vítima)
- RFIDSensor
- TouchSensor (sensor de toque)

2.11 Combinações de robôs

As equipes podem usar qualquer combinação dos robôs representados nas figuras(FIGURAS. . . ,) abaixo:

a) Real ATRVJr

b) Simulated ATRVJr

Figura 2.2: ATRVJr

a) Real HMMWV

b) Simulated HMMWV

Figura 2.3: Hummer

a) Real P2AT

b) Simulated P2AT

Figura 2.4: P2AT

a) Real P2DX

b) Simulated P2DX

Figura 2.5: P2DX

a) Real Talon

b) Simulated Talon

Figura 2.6: Talon

a) Real TeleMax b) Simulated Telemax

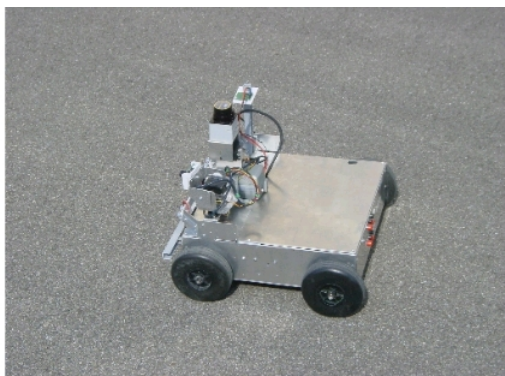
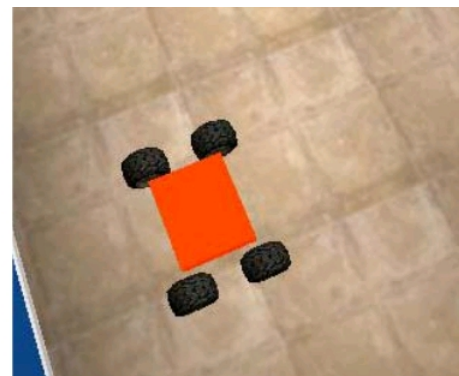
Figura 2.7: Telemax

a) Real Zerg b) Simulated Zerg

Figura 2.8: Zerg

3 *Proposta*

O objetivo deste capítulo é identificar a proposta propriamente dita, deixando claras as necessidades do controlador e definindo uma arquitetura. A primeira parte do trabalho fica definida como sendo um processo de reescrita do que já foi feito, um refactoring do código criado pelo ITA.

3.1 Arquitetura do Controlador

Na arquitetura proposta em conjunto com as demais universidades envolvidas, foi decidido que o controlador será desenvolvido em C++, pelas funções e recursos providos por essa linguagem. Decidiu-se também que cada robô deve ser considerado um processo independente dos outros robôs, e que para conectar-se ao servidor, exportará uma API de controle via socket TCP. Essa arquitetura traz algumas vantagens, como por exemplo, a escalabilidade, pois haveria a capacidade deste sistema em suportar um aumento do número de robôs, já que cada robô pode ser executado em uma maquina diferente. Além disso,fica fácil fazer uma analogia com um robô real, considerando que é uma característica destes dispositivos serem autônomos.

3.2 Requisitos do Controlador

O controlador deve obedecer aos seguintes requisitos:

- Ser multiplataforma - Para desenvolver seus algoritmos, os usuários não devem preocupar-se com requisitos de sistema operacional.
- Extensível – Desta forma, o usuário pode criar seus próprios algoritmos, além de aproveitar as demais funcionalidades do controlador.
- Escalável – O número de processos que podem ser rodados ao mesmo tempo pode ser aumentado e o sistema deve adaptar-se a esta mudança.

- Fácil para testar novos algoritmos – O controlador também servirá de ferramenta didática, portanto deve propiciar grande facilidade para testar algoritmos que a extendam.

Características dos robôs:

- Os robôs devem permitir a adição de plugins, pois dessa forma códigos e algoritmos podem ser acrescentados de forma simples, tornando-o extensível. Uma interface bem definida é imprescindível para cada ação que robô tomar, pois isso torna fácil e rápido testar novos algoritmos como os de SLAM.
- Através da API do robô deve ser possível tanto obter os dados dos sensores quanto controlá-lo.
- Deve ser possível executar o robô através da linha de comando, passando como argumentos os seus parâmetros de configuração. Algo como: ./robot –usarsim-address 192.168.0.1 –usarsim-port 3000 –initial-position 0,0,0 –port 12345 –slam NomeDoAlgoritmo ...

  Isso torna desnecessária a abertura da interface para executar um robô. Além de propiciar uma forma simples tanto de configuração quanto de disparo do robô. É necessário, apenas, montar a linha de comando e criar o novo processo.

3.3 Interface

Para a implementação da interface foi decidido em conjunto com as demais universidades envolvidas que a melhor framework gráfica é a WxWidgets(WXWIDGETS,), pelo fato desta ser multiplataforma e adquirir o look and feel do sistema onde está sendo utilizada. É uma ferramenta bastante completa escrita em C++, mesma linguagem em que será escrito o controlador. Usando esta framework é mantido o objetivo inicial de obtermos um controlador bem genérico e completo.

3.4 Projeto no tocante da UFSC

Em discussão com as universidades ficou decidido que cada entidade deveria sugerir qual parte do projeto gostaria de estudar e implementar. Levando-se em conta este fato, decidiu-se que na UFSC trabalharemos com a parte de SLAM(Simultaneous Localization And Mapping) do controlador. Foi decidido por esta parte, pelo fato deste projeto ser voltado para a Inteligência

Artificial, e acredita-se que neste ponto fica clara a necessidade de algorítmos que se enquadram nessa área de estudo. Em outras palvras, cabe aos graduandos envolvidos estudar e implementar técnicas que serão utilizadas pelos robôs e veículos autônomos para que estes possam construir um mapa dentro do ambiente, até então desconhecido, sem que percam o controle da sua posição atual.

3.5 Definições pendentes

Definir as arquiteturas dos dois módulos, tanto Robot quanto User Interface. Definir os diagramas UML, deixando clara a área em que cada uma das universidades irá atuar para então começar a fase de implementação.

Referências Bibliográficas

VIRTUAL Robots Competitions Rules Figura SW/HW configuration. Disponível em: <http://www.robocuprescue.org/wiki/images/Rules2007V5.pdf>.

RICH, E.; KNIGHT, K. *Inteligencia artificial*. [S.l.]: McGraw-Hill, 1994.

PIRES, J. N. *Robótica Das Máquinas Gregas à Moderna Robótica Industrial*. Jul 2002. Disponível em: <http://robotics.dem.uc.pt/norberto/nova/pdfs/gregosxxi.pdf>.

ROBOCUP. Disponível em: <http://www.robocup.org/>.

VIRTUAL Robots Competition. Disponível em: <http://www.robocuprescue.org/rescuerobots.html>.

USARSIM. Disponível em: <http://sourceforge.net/projects/usarsim/>.

VIRTUAL Robots Competition Rules. Disponível em: <http://www.robocuprescue.org/wiki/images/Rules2007V5.pdf>.

MOAST. Disponível em: <http://sourceforge.net/projects/moast/>.

ROBOCUP Rescue Simulation Agent Competition. Disponível em: <http://www.robocuprescue.org/wiki/index.php?title=RSL2007>.

FIGURAS Robôs USARSim. Disponível em: <http://sourceforge.net/project/showfiles.php?group$_{i}d = 145394$>.

WXWIDGETS. Disponível em: <http://www.wxwidgets.org/>.